

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 June 2001 (07.06.2001)

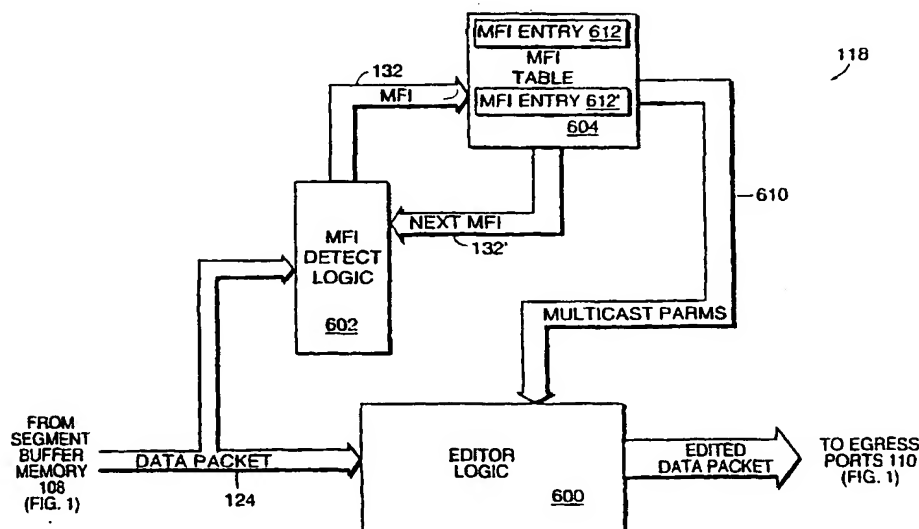
PCT

(10) International Publication Number
WO 01/41364 A2

- (51) International Patent Classification⁷: **H04L 12/18**, 29/06, 12/46
- (21) International Application Number: PCT/CA00/01420
- (22) International Filing Date:
30 November 2000 (30.11.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/453,344 1 December 1999 (01.12.1999) US
- (71) Applicant (for all designated States except US): **MOSAID TECHNOLOGIES INCORPORATED** [CA/CA]; 11 Hines Road, Kanata, Ontario K2K 2X1 (CA).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **BROWN, David, A.** [CA/CA]; 110 Allissia Crescent, Carp, Ontario K0A 1L0 (CA).
- (74) Agent: **PILLAY, Kevin**; Fasken Martineau DuMoulin LLP, Suite 4200, Box 20, Toronto-Dominion Centre, Toronto Dominion Bank Tower, Toronto, Ontario M5K 1N6 (CA).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— Without international search report and to be republished upon receipt of that report.

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR WIRE SPEED IP MULTICAST FORWARDING



(57) Abstract: In a switch including a plurality of ports, an IP Multicast packet arriving on an ingress port a copy of the receiver packet is forwarded to each member of the IP Multicast group at wire-speed. The packet is bridged once to a given egress port and may be routed multiple times out of the egress port. If multiple subnets exist on an egress port, each subnet that requires a copy of the packet will receive the packet with its VLAN ID included in the packet. The received IP Multicast packet for an IP Multicast group is stored in memory, a pointer to the location of the packet in memory is stored for each port to which a copy of the packet is to be forwarded. An IP Multicast forwarding entry is provided for the IP Multicast group. The forwarding entry includes a modification entry for each packet to be forwarded to the IP Multicast group. A copy of the stored packet is modified dependent on the modification entry and forwarded in the next available port cycle for the port.

WO 01/41364 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR WIRE SPEED IP MULTICAST FORWARDING

BACKGROUND OF THE INVENTION

5 In a computer network, a networking switch receives a data packet at an ingress port connected to the switch and forwards the data packet to an egress port connected to the switch. The switch determines the egress port to which the data packet is forwarded dependent on a destination address included in the data packet received at the ingress port.

10 A destination node connected to an egress port may be a member of an IP Multicast group. A destination node joins an IP Multicast group in order to receive IP Multicast data such as a real-time video. A switch receiving an IP Multicast data packet at an ingress port must forward a copy of the received IP Multicast data packet to all members of the IP Multicast Group connected to the switch.

15 By transmitting one IP Multicast data packet to the ingress port instead of multiple unicast data packets, one for each member of the IP Multicast group, the number of data packets transmitted to the switch is reduced. However, the time to process the IP Multicast data packet in the switch is increased because the switch must determine to which egress ports the IP Multicast data packet is to be forwarded, and the number of copies to forward to each egress port.

20 The switch must forward a copy of the IP Multicast data packet to each member of the IP Multicast group. Thus, multiple copies of the IP Multicast data packet are forwarded to an egress port if multiple members of the IP Multicast group are connected to the egress port. A copy of the IP Multicast data packet is modified
25 by the networking switch before being forwarded to a member of the IP Multicast group.

Typically, the forwarding of an IP Multicast data packet through a switch to an egress port, is performed by a host processor in the networking switch. Host processor based forwarding is slow.

SUMMARY OF THE INVENTION

5 A copy of an IP Multicast data packet received at an ingress port of a networking switch is forwarded at wire speed to an egress port. The IP Multicast data packet received at the ingress port is stored in memory. A copy of the stored IP Multicast data packet is forwarded to egress port forwarding logic. The egress port forwarding logic includes a modification entry for each member of the IP Multicast group associated with the IP Multicast data packet. Upon receiving a copy of an IP
10 Multicast data packet from memory, the egress port forwarding logic modifies the copy of the IP Multicast data packet dependent on a modification entry associated with the egress port to which the IP Multicast data packet is to be forwarded.

15 Ingress port forwarding logic in the switch generates a multicast forward vector for the IP Multicast data packet. The Multicast forward vector indicates to which port the stored IP Multicast data packet is to be forwarded.

20 Port queue logic in the switch stores a pointer to the location of the IP Multicast data packet in memory on a port queue dependent on the multicast forward vector. The port queue logic removes the pointer after the IP Multicast data packet has been forwarded to all members of the IP Multicast group connected to the egress port associated with the port queue.

25 The modification entry may include an identification for a Virtual LAN, a pointer to a next modification entry, a tag indicator and a forward type entry. The pointer to a next modification entry is an index to a next modification entry for modifying the next IP Multicast data packet for the port.

The egress port forwarding logic may include a Virtual LAN identification table and an intermediate LAN identification table. The intermediate LAN

identification table includes an index to a location in the virtual LAN table and the internal LAN table stores a Virtual LAN identification.

BRIEF DESCRIPTION OF THE DRAWINGS

5 The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon
10 illustrating the principles of the invention.

Fig. 1 illustrates a network switch including IP Multicast forwarding logic for forwarding an IP Multicast packet received at an ingress port to members of an IP Multicast group connected to the switch according to the principles of the present invention;

15 Fig. 2 is a timing diagram illustrating the forwarding of IP Multicast data packets by the switch shown in Fig. 1 to members of the IP Multicast group shown in Fig. 1;

Fig. 3A illustrates a prior art data packet which may be received on an ingress port connected to a switch;

20 Fig. 3B illustrates a prior art Ethernet Data link layer (L2) header which may be included in the data packet shown in Fig. 3A;

Fig. 3C illustrates a prior art Internet Protocol (Network layer (L3)) header which may be included in the data packet shown in Fig. 3A;

Fig. 3D illustrates the prior art Tag Control Information shown in Fig. 3B;

25 Fig. 4 illustrates the ingress IP Multicast forwarding logic shown in Fig. 1;

Figs. 5 is a flowchart of the method for processing an IP Multicast data packet received on an ingress port;

Fig. 6 illustrates the egress multicast forwarding logic shown in Fig. 1;

Fig. 7 illustrates the Multicast Forwarding Index table shown in Fig. 6; and

30 Fig. 8 is a flowchart of the steps performed in the egress multicast forwarding logic shown in Fig. 6 for modifying the IP Multicast data packet.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 illustrates the members of an IP Multicast group connected to a network switch 100. The members of the IP Multicast group include destination nodes 104a-f connected to egress ports 110a-c. Source node 102 connected to an ingress port 108 is the source of IP Multicast data packets for the IP Multicast group. For example, the source node 102 may be transmitting a video to all members of the IP Multicast group. In an alternative embodiment source node 102 may be a member of the IP Multicast group. The source node 102 forwards an IP Multicast data packet 106 to the switch 100. The switch 100 forwards a copy of the IP Multicast data packet 106 to each member of the IP Multicast group, that is, destination nodes 104a-f.

The copy of the IP Multicast data packet 106 may be modified by the switch 100 before it is forwarded to each destination node 104. The copy of the received IP Multicast data packet 106 is modified dependent on a modification entry stored in the switch 100. The modification and forwarding of the copy of the received IP Multicast data packet is performed by the switch at wire-speed, that is, the modification is performed without host intervention and the egress Modified IP Multicast data packet 112 is forwarded to a destination node 104 in the next available port cycle for the egress port 110 to which the destination node 104 is connected.

As an IP Multicast data packet 106 is received on an ingress port 108, the data packet processing logic 128 extracts information from the header of the IP Multicast data packet 106 and forwards the extracted information to the ingress Multicast forwarding logic 114. The ingress Multicast forwarding logic 114 determines to which egress ports 110 the destination nodes 104 in the IP Multicast group are connected and forwards a Multicast forward vector 120 to the egress port-q logic 134 indicating the egress ports 110 to which the IP Multicast data packet is to be forwarded. The ingress Multicast forwarding logic 114 also forwards a Multicast Forwarding Index ("MFI") 132 corresponding to the IP Multicast group to

which the IP Multicast data packet 106 is to be forwarded, to the data packet processing logic 128. The MFI 132 is an index to a modification entry for each member of the IP Multicast group.

5 The data packet processing logic 128 forwards the received IP Multicast data packet 106 and the MFI 132 to memory 116. The ingress modified IP Multicast data packet 126 including the MFI 132 is stored in memory 116. The address of the ingress modified IP Multicast data packet 126 is stored in a port queue (not shown) for each egress port 110 on which the IP Multicast data packet 106 is to be forwarded. Port queues are described in co-pending U.S. Application Serial No. 10 09/386,589, filed on August 31, 1999 entitled "Method and Apparatus for an Interleaved Non-Blocking Packet Buffer" by David A. Brown, the entire teachings of which are incorporated herein by reference.

15 The MFI 132 stored in the ingress modified IP Multicast data packet 126 is used to access a stored modification entry associated with each member of the IP Multicast group. The egress Multicast forwarding logic 118 modifies a copy of the ingress modified IP Multicast data packet 126 received from memory 116 using the contents of the stored modification entry for the member and forwards an egress modified IP Multicast data packet 112 to each member of the IP Multicast group.

20 By storing a modification entry for each member of the IP Multicast group, no host processing is required to modify and forward a copy of the IP Multicast data packet 106 to all members of the IP Multicast group. Thus, the IP Multicast data packet is forwarded at wire speed, that is, to a member of the IP Multicast group connected to a port in the next available port cycle for the port. The modification entry associated with each destination node 104 is stored in the egress Multicast forwarding logic 118 and is accessed using the MFI 132 associated with the IP 25 Multicast Group forwarded from the Ingress Multicast forwarding logic 114. The MFI is stored in the ingress modified Multicast data packet 126 in memory 116.

As shown in Figure 1, the members of the IP Multicast group include destination nodes Da-f 104a-f. Destination nodes Da-c 104a-c are connected to egress port_1 110a, destination nodes Dd-e 104d-e are connected to egress_port 2 110b and destination node Df 104f is connected to egress port_3 100c. All the destination nodes 104a-f in the embodiment shown are terminating nodes but in an alternative embodiment any one of the destination nodes 104a-f shown may be the ingress port of another switch and a copy of the IP Multicast data packet may be forwarded through the other switch to members of the IP Multicast group connected to the other switch's egress ports. Also, there may be multiple VLANs connected to the egress ports of the other switch's egress ports, with each VLAN requiring a modified copy of the IP Multicast data packet.

A data packet 106 transmitted by the source node Sa 102 is received by the network switch 100 at ingress port_1 108. The data packet 106 includes a destination address. If the destination address is an IP Multicast address, the switch 100 forwards a modified copy of the IP Multicast data packet 112 to all members of the IP Multicast group associated with the IP Multicast address connected to the switch 100.

For the IP Multicast group shown, the switch 100 forwards six egress modified IP Multicast data packets 112a-f to egress ports 1-3 110a-c. An egress modified IP Multicast data packet 112 is forwarded to the egress port 110 in the next available port cycle for the egress port 110 after the IP Multicast data packet 106 has been received at the ingress port 102.

Three egress modified IP Multicast data packets 112a-c are forwarded to egress port_1 110A, one to each of destination nodes Da-c 104a-c. Two egress modified IP Multicast data packet 112d-e are forwarded to egress port_1 110b, one to each of destination nodes Dd-e 104d-e. One egress modified IP Multicast data packet 112f is forwarded to egress port_2 110c to destination node Df 104f.

Upon identifying the received IP Multicast data packet 106, the data packet processing logic 128 forwards a portion of the header of the IP Multicast data packet 130 to ingress Multicast forwarding logic 114. The ingress Multicast forwarding logic 114 determines the egress ports_1-3 110a-c to which a copy of the IP Multicast data packet 106 is to be forwarded and generates a Multicast forward vector 120. The Multicast forward vector 120 is forwarded to the egress port_q logic 134. The ingress Multicast forwarding logic 114 selects a stored Multicast Forwarding Index ("MFI") 132 dependent on the portion of the header of the IP Multicast data packet 130 and forwards the MFI 132 to the data packet processing logic 128. The data packet processing logic 128 stores the MFI 132 with the received IP Multicast data packet 106 in the ingress modified IP Multicast data packet 126 and forwards the ingress modified IP Multicast data packet 126 on data-in 122 to memory 116. The ingress modified IP Multicast data packet 126 is stored in memory 116.

At the next port cycle for each of the egress ports_1-3 110a-c, the ingress modified IP Multicast data packet 126 is forwarded from memory 116 on data-out 124 to the egress Multicast forwarding logic 118 dependent on the address 136 forwarded from egress port_q logic 134. The egress Multicast forwarding logic 118 modifies the copy of the ingress modified data packet 126 dependent on the MFI 132 included in the ingress modified data packet 126. The egress modified data packets 112a-f are forwarded on the respective egress ports_1-3 110a-c. After all the egress modified data packets 112a-f have been forwarded, the ingress modified data packet 126 stored in memory 116 may be overwritten.

Fig. 2 is a timing diagram illustrating wire speed forwarding of an IP Multicast data packet 106 through the switch 100 to members of the IP Multicast group shown in Fig. 1. Fig. 2 is described in conjunction with Fig. 1. The IP Multicast group shown in Fig. 1 includes six members, Da-Df 104a-f connected to the switch 100. The switch 100 includes four ports, ingress_port 1 108 and egress ports 1-3 110a-b. However, the invention is not limited to four ports or six IP Multicast group members as shown. It is well-known to those skilled in the art that

the configuration in Fig. 1 may be expanded to include greater than four ports, and an IP Multicast group with greater than six members.

In the timing diagram, it is assumed that each IP Multicast data packet 106 received at ingress port_1 108 is the minimum data packet size, for example, 64 bytes, for an Ethernet data packet because the memory 116 is 64 bytes wide. However, the invention is not limited to a 64 byte packet size, the number of bytes in the data packet may be greater than the width of the memory 116.

It is also assumed that the 64 byte IP Multicast data packet 106 is received at an ingress port 108 (Fig.1) during a data segment time slot 200. The data segment time slot 200 as shown includes 4 port cycle time slots 202 one for each port connected to the switch 100 because the switch shown in Fig. 1 includes four ports connected to the switch 100, that is, one ingress port 108 and three egress ports 110a-c. The maximum number of ports supported by the switch 100 is dependent on the minimum port cycle time slot 202 and the maximum data segment time slot 200 and thus is not limited to the four ports shown in Fig. 1. In a switch 100 with greater than four ports the data segment time slot 200 would include a port cycle time slot 202 for each port connected to the switch 100. For example, for an Ethernet data packet, the minimum data packet size is 64 bytes. Thus, the maximum data segment time slot 200 is the time it takes to receive 64 bytes at an ingress port and the minimum port cycle time slot 202 is dependent on the time to access the 64 bytes from memory 116 (Fig. 1). Data segment time slots and port cycle time slots are described in co-pending U.S. Application Serial No. 09/386,589, filed on August 31, 1999 entitled "Method and Apparatus for an Interleaved Non-Blocking Packet Buffer" by David A. Brown, the entire teachings of which are incorporated herein by reference.

A copy of an IP Multicast data packet 106 arriving at ingress port_1 108 is forwarded in the next available port cycle time slot 202 to each member of the IP Multicast group, that is, it is forwarded at wire speed. If only one member of the IP Multicast group is connected to the port the copy of the IP Multicast data packet is

forwarded in the next port cycle time slot 202. If more than one member of the IP Multicast group is connected to a port, the next IP Multicast data packet arriving at the ingress port 108 is forwarded after the previous IP Multicast data packet has been forwarded to all members of the IP Multicast group connected to the port.

5 The timing diagram shows four IP Multicast data packets arriving at ingress port_1 108 from source node-a 102 in data segment time slots 200a-d. The first IP Multicast data packet arrives in data segment time slot 200a. In the second data segment time slot 200b, the IP Multicast data packet is forwarded to egress port_1 110a in port cycle 202b, to egress port_2 110b in port cycle 202c and to egress
10 port_3 110c in port cycle 202c. As the first IP Multicast data packet is being forwarded, the second IP Multicast data packet is being received during data segment time slot 200b.

At time 230, ingress modified IP Multicast data packet _1 126 (Fig. 1) is stored in memory 116 (Fig. 1) and ingress multicast forwarding logic 114 (Fig. 1)
15 has forwarded the multicast forward vector 120 (Fig. 1) for IP Multicast data packet _1 106 (Fig. 1) to egress multicast forwarding logic 118 (Fig. 1).

At time 206, the start of port cycle time slot 202b, a copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is forwarded from memory 116 (Fig. 1) on
20 data-out 124 (Fig. 1) to egress Multicast forwarding logic 118 (Fig. 1) dependent on the address 136 forwarded from egress port_q logic 134 (Fig. 1). Egress Multicast forwarding logic 118 (Fig. 1) modifies the received copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) dependent on the MFI 132 (Fig. 1). Egress modified IP Multicast data packet_1 112A is forwarded through egress port _1 110A
25 to destination node Da 104a.

At time 208, the start of port cycle time slot 202c, and at time 210, the start of port cycle time slot 202d, a copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is forwarded from memory 116 (Fig. 1) on data-out 124 (Fig. 1) dependent on the address 136 (Fig. 1) forwarded from egress port_q logic 134 (Fig.

1). The received copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is modified by egress Multicast forwarding logic 118 dependent on the associated MFI 132 (Fig. 1) and forwarded through egress port 2 110b to destination node Dd 104d and egress port 3 110c to destination node Df 104f.

5 At time 212, a copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is modified and forwarded to destination node Db 104b (Fig. 1) and at time 218 a copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is modified and forwarded to destination node Dc 104c connected to egress port _2 110b (Fig. 1). At time 214, a copy of ingress modified IP Multicast data packet _1 126 (Fig. 1) is modified and forwarded to destination node Dd 104d connected to egress port _1 110b (Fig. 1).

10 At time 232, ingress modified IP Multicast data packet _2 126 (Fig. 1) is stored in memory 116 (Fig. 1) and ingress Multicast forwarding logic 114 (Fig. 1) has forwarded the Multicast forward vector 120 (Fig. 1) for IP Multicast data packet _2 106 (Fig. 1) to egress Multicast forwarding logic 118 (Fig. 1). At time 234, ingress modified IP Multicast data packet _3 126 (Fig. 1) is stored in memory 116 (Fig. 1). At time 236, ingress modified IP Multicast data packet _4 126 (Fig. 1) is stored in memory 116 (Fig. 1).

15 Only one egress modified IP Multicast data packet _1 112 (Fig. 1) is forwarded to egress port _3 110c because only one member of the IP Multicast group, that is, destination node 104f is connected to egress port _3 110c. Thus, at time 216, in port cycle time slot 202d, egress modified IP Multicast data packet _2 112f is forwarded to egress port _3 110c. At time 222, in port time cycle 202d, egress modified IP Multicast data packet _3 112f is forwarded to egress port _3 110c. At time 228, in port cycle time slot 3 202d, egress modified IP Multicast data packet _4 112f is forwarded to egress port _3 110c (Fig. 1).

20 Thus, IP Multicast data packet _1 106 (Fig. 1) received at ingress port _0 108 (Fig. 1), is forwarded through the switch 100 (Fig. 1) at wire speed without host

processing in the next available port cycle time slot 202 to destination nodes 104a-f (Fig. 1) connected to egress ports _1-3 110a-c (Fig. 1).

Egress modified IP Multicast data packet_2 112 (Fig. 1) is forwarded on egress port_2 110b to destination node De 104e and destination node Df 104f. At time 220, egress modified IP Multicast data packet_2 112d is forwarded to destination node Dd 104d connected to egress port_2 110b. At time 226, egress modified IP Multicast data packet_2 112e (Fig. 1) is forwarded to destination node De 104e connected to egress port_2 110b.

At time 224, after the last egress modified IP Multicast data packet_1 112 (Fig. 1) has been forwarded, the location in memory storing ingress modified IP Multicast data packet _1 126 (Fig. 1) may be used to store another data packet received at the ingress port 108 (Fig. 1). Thus, the ingress modified IP Multicast data packet 126 (Fig. 1) received at the ingress port 108 (Fig. 1) is stored in memory 116 (Fig. 1) until egress modified IP Multicast data packet 112 (Fig. 1) has been forwarded to all members of the IP Multicast group.

To determine if all egress modified IP Multicast data packets 112a-f (Fig. 1) have been forwarded, the number of port queues (not shown) that the ingress modified IP Multicast data packet 126 (Fig. 1) has been queued on associated with each ingress modified IP Multicast data packet 126 (Fig. 1) is stored in a port queue counter register (not shown). The counter register is decremented each time all egress modified IP Multicast data packets 112 (Fig. 1) associated with an egress port 110 (Fig. 1) have been forwarded to the egress port 110 (Fig. 1). For example, the port queue count is decremented after one egress modified IP Multicast data packet has been forwarded on egress port_3 110c. The number of port queues is determined from the Multicast forward vector 120 (Fig. 1).

The prior art data packet received on ingress port 108 (Fig. 1) is described in conjunction with Figs. 3A-3D in order to describe the modifications that the forwarding logic 114, 118 (Fig. 1) and data packet processing logic 126 (Fig. 1) may

perform on an IP Multicast packet 106 (Fig. 1) before egress modified IP Multicast data packets 112a-f (Fig. 1) are forwarded to egress ports 110a-c.

Fig. 3A illustrates a prior art data packet which may be received on an ingress port 108 (Fig. 1) connected to the switch 100 (Fig. 1). Fig. 3B illustrates a prior art Ethernet header which may be included in the data packet shown in Fig. 3A. Fig. 3C illustrates a prior art Internet Protocol header which may be included in the data packet shown in Fig. 3A. Fig. 3D illustrates the prior art Tag Control Information field 318B in the Ethernet header shown in Fig. 3B.

Fig. 3A shows a prior art data packet 300. The data packet 300 includes a data field 310 and headers for networking layers 302, 304, 306, 308. Headers for four of the layers in the OSI model are shown, the physical layer (L1) header 302, the data link layer (L2) header 304, the networking layer (L3) header 306 and the transport layer (L4) header 308. For example, the data link layer (L2) header 304 may be Ethernet and the networking layer (L3) header 306 may be Internet Protocol ("IP"). The data packet 300 also includes a checksum 312.

Fig. 3B illustrates the format of an Ethernet data link (L2) header 304. The Ethernet data link (L2) header 304 includes a device address for the destination node 104 (Fig. 1), that is, the destination address 314, and a device address for the source node 102 (Fig. 1), that is, the source address 316, an optional Virtual Local Area Network Identification ("VLAN ID") field 318 and a length/type field 320. The VLAN ID 318 includes a Tag Protocol Identifier ("TPI") field 318A and a Tag Control Information ("TCI") field 318B. The VLAN ID field 318 provides support for VLAN switching based on IEEE 802.1Q tagging and IEEE 802.ID 1988 (802.1p) priority levels.

FIG. 3C illustrates the format of an IP network layer (L3) header 306. The IP network layer (L3) header 306 includes a network address for the source node 102 (Fig. 1), that is the source IP address 344, and a network address for the destination node 104 (Fig. 1), that is, the IP destination address 346. Other fields in

the IP network layer header 306 include Version 322, HLEN 324, Type of Service ("TOS") 326, Total Length 328, Identification 330, Flags 332, Fragment Offset 334, Time to Live ("TTL") 336, Protocol 340, Header Checksum 342, Options 348 and pad 350.

5 An IP Multicast data packet 106 (Fig. 1) received at the ingress port 108 (Fig. 1) connected to the switch 100 (Fig. 1) is bridged to one or more egress ports 110a-c (Fig. 1) dependent on the destination address 314 (Fig. 3B) stored in the Ethernet data link (L2) header 300 (Fig. 3A) or is routed to one or more egress ports 110 (Fig. 1) dependent on the IP destination address 346 stored the IP network layer (L3) header 306.

Fig. 3D illustrates the prior art TCI field 318b shown in Fig. 3B. The TCI field 318b includes a 3-bit user priority field 352, a 1-bit CFI field 354 and a 12-bit Virtual Local Area Network ("VLAN") identifier 356.

15 Returning to Fig. 1, the data packet processing logic 128 determines if a data packet received at ingress port_1 108 is an IP Multicast data packet 106 by examining the network layer (L3) header 306 (Fig. 3A) included in the received data packet. The network layer (L3) header 306 (Fig. 3A) is an Internet Protocol Version 4 ("IPv4") header if the protocol field 340 (Fig. 3C) is IP, the version field 322 (Fig. 3C) is IPv4 and the header checksum 342 (Fig. 3C) is valid. The IP data packet is an IP Multicast data packet 106 if the IP destination address 346 (Fig. 3C) is class D, that is, the high order four bits of the IP destination address 346 (Fig. 3C) are set to 1110.

25 If the data packet processing logic 128 determines that the data packet is an IP Multicast data packet 106, a portion of the headers of the IP Multicast data packet 130 including the IP Destination address 346 (Fig. 3C) and IP Source address 344 (Fig. 3C) in the IP networking layer (L3) header 306 (Fig. 3C), the VLAN ID 318 (Fig. 3B) in the Ethernet data link (L2) header 304 (Fig. 3B), and the ingress port

number corresponding to the ingress port 108 (Fig. 1) are forwarded to the Ingress Multicast Forwarding Logic 114.

Fig. 4 illustrates the ingress Multicast forwarding logic 114 shown in Fig. 1. The ingress Multicast forwarding logic 114 includes a Content Addressable Memory ("CAM") 400, a route mapper 402, a route table 404 and combine vector logic 406. The route table 404 includes a Multicast forwarding entry 422 corresponding to the IP Multicast Group included in the IP Multicast data packet 106 (Fig. 1) received at ingress port _1 108 (Fig. 1). The CAM 400 and the route mapper 402 select a Multicast forwarding entry 422 dependent on the portion of the headers of the IP Multicast data packet 130 (Fig. 1) and the port number for ingress port _1 108 (Fig. 1) forwarded from the data packet processing logic 128 (Fig. 1).

In the implementation of the Ingress Multicast forwarding logic 114 (Fig. 1) described in conjunction with Fig. 4, the CAM 400 is not sufficiently wide to store the IP Destination Address 346 (Fig. 3C), IP Source Address 344 (Fig. 3C), ingress port number and VLAN ID 318 (Fig. 3B). Thus, the route mapper 402 is implemented to increase the width of the CAM 400. However, it is well known by those skilled in the art that the CAM 400 may be expanded to include the Route Mapper 402.

The search for the Multicast Forwarding entry 422 in the Route Table 404 corresponding to the IP Multicast packet 106 is performed in two stages. First, the CAM 400 is searched for an entry dependent on the IP destination address 346 (Fig. 3C) and the IP Source Address 344 (Fig. 3C) fields in the IP network layer (L3) header 306 (Fig. 3C). If a matching entry is found in the CAM 400, a match address 410 is forwarded to the route mapper 402. The route mapper 402 forwards a Multicast Path Code ("MPC") 422 corresponding to the match address 410 to the CAM 400.

Second, the CAM 400 is searched for an entry dependent on the MPC 422, the VLAN ID 318 (Fig. 3B) and the ingress port number. If a matching entry is

found in the CAM 400, a match address 410 is forwarded to the route mapper 402. The route mapper 402 forwards an Associated Data Index ("ADI") corresponding to the match address 410 to the route table 404. An IP Multicast forwarding entry 422 is stored at a location in the route table 404 identified by the ADI 412. The IP
5 Multicast forwarding entry 422 identifies the IP Multicast group to which the IP Multicast data packet 106 (Fig. 1) is to be forwarded by the switch 100 (Fig. 1).

The CAM 400 is searched with the IP source address 344 (Fig. 3C), the IP destination address 346 (Fig. 3C), the VLAN ID 318 (Fig. 3B) and the ingress port number 108 (Fig. 1) in order to prevent duplicating forwarding of the IP Multicast data packet. For example, if a source node is connected to more than one port, that is, destination node x is connected to port y and port x and an IP Multicast data packet is received from destination node x on port y, the data packet is forwarded to all members of the IP Multicast group. However, if the IP Multicast data packet is received from destination address x on port z it is discarded because destination x is
10 connected to more than one port and thus the IP Multicast data packet received on port z is a duplicate of the IP Multicast data packet received on port y.
15

Each Multicast forwarding entry 422 in the Route Table 404 includes a bridge vector 416, a route vector 414 and a Multicast Forwarding Index 132 ("MFI"). The bridge vector 416 and route vector 414 are forwarded to combine vectors logic 406. The MFI 132 is forwarded to the data packet processing logic 128 (Fig. 1).
20

The combine vector logic 406 generates a Multicast forward vector 120 by performing a logical OR function on the route vector 414 and the bridge vector 416. The Multicast forward vector 120 is forwarded to the egress port_q logic 134 (Fig. 1). The Multicast forward vector 120 selects one or more egress ports to which the IP Multicast data packet 106 (Fig. 1) is to be forwarded.
25

The first three bytes of the destination address 314 (Fig. 3B) in the Ethernet data link (L2) header 304 (Fig. 3B) of the IP Multicast data packet 106 (Fig. 1)

received at ingress port_1 108 are set to "0x01-00-53". Before the IP Multicast data packet 106 is forwarded on data-in 122 (Fig. 1) to memory 116 (Fig. 1), the data packet processing logic 128 (Fig. 1) writes the MFI 132 forwarded from ingress Multicast logic 114 (Fig. 1) into the second and third bytes of the destination address 314 (Fig. 3B) as follows: "0x01-0M-F1". The ingress modified IP Multicast data packet 126 (Fig. 1) is stored in memory 116 (Fig. 1). Egress Multicast forwarding logic 118 (Fig. 1) uses the MFI 132 included in the ingress modified IP Multicast data packet 126 (Fig. 1) to modify the ingress modified IP Multicast data packet 126 (Fig. 1) before forwarding an egress modified IP Multicast data packet 112 (Fig. 1) to an egress port 110 (Fig. 1).

The IP Multicast forward vector 120 determines to which egress ports 110a-c (Fig. 1) the IP Multicast data packet 106 (Fig. 1) is to be forwarded. The location of the ingress modified Multicast data packet 126 (Fig. 1) in memory 116 (Fig. 1) is stored on an egress port queue (not shown) in egress port_q logic 134 (Fig. 1) dependent on the state of a corresponding egress port queue bit (not shown) implemented in the ingress multicast forwarding logic 114 shown in Fig. 5 in the Multicast forward vector 420.

Fig. 5 is a flowchart of the method implemented in the ingress multicast forwarding logic 114 shown in Fig. 4 for processing an IP Multicast data packet 106 (Fig. 1) received at an ingress port 108 (Fig. 1). The method is described in conjunction with Figs. 1-4.

At step 500, the ingress multicast forwarding logic 114 (Fig. 1) receives a portion of the IP Multicast data packet 130 from the data packet processing logic 128 (Fig. 1). Processing continues with step 502.

At step 502, the ingress Multicast forwarding logic 114 (Fig. 4) initializes the route vector 414 (Fig. 4) and the bridge vector 416 (Fig. 4) in the combine vectors logic 406 (Fig. 4).

At step 504, a search is performed in the CAM 400 (Fig. 4) for an entry matching the IP destination address 346 (Fig. 3C) and the IP source address 344 (Fig. 3C) forwarded to the CAM 400 (Fig. 4) from the data packet processing logic 128 (Fig. 1). If a match address 410 (Fig. 4) is found in the CAM 400 (Fig. 4),
5 processing continues with step 506. If not, processing continues with step 518.

At step 506, the match address 410 (Fig. 4) is forwarded to the route mapper 402 (Fig. 4). The route mapper 402 (Fig. 4) forwards a Multicast Path Code ("MPC") 422 (Fig. 4) corresponding to the match address 410 (Fig. 4) to the CAM 400 (Fig. 4). Processing continues with step 508.

10 At step 508, a second search is performed in the CAM 400 (Fig. 4) for an entry dependent on the MPC 422, and the VLAN ID 318 (Fig. 3B) from the Ethernet data link (L2) header 304 (Fig. 3B) and ingress port number. If a match address 410 (Fig. 4) is found in the CAM 400 (Fig. 4) processing continues with step 510. If not, processing continues with step 518.

15 At step 510, the match address 410 (Fig. 4) is forwarded to the route mapper 402 (Fig. 4). The route mapper 402 (Fig. 4) forwards an ADI corresponding to the match address 410 (Fig. 4) to the route table 404. The IP Multicast forwarding entry 422 (Fig. 4) for the IP Multicast data packet 106 (Fig. 1) is stored at a location
20 in the route table 404 identified by the ADI 412. Processing continues with step 512.

At step 512, the MFI 132 (Fig. 4) stored in the Multicast Forwarding entry 422 (Fig. 4) is forwarded to data packet processing logic 128 (Fig. 1) and the bridge vector 416 (Fig. 4) stored in the Multicast forwarding entry 422 (Fig. 4) is forwarded
25 to the combine vectors logic 406 (Fig. 4). Processing continues with step 514.

At step 514, if the TTL field 336 (Fig. 3C) in the IP network layer (L3) header 306 (Fig. 3C) in the IP Multicast data packet 106 (Fig. 1) forwarded to the ingress Multicast forwarding logic 114 (Fig. 1) on the portion of the IP Multicast

data packet 130 (Fig. 1) is greater than one processing continues with step 516. If not, processing continues with step 518.

At step 516, the route vector 414 (Fig. 4) stored in the Multicast forwarding entry 422 (Fig. 4) is forwarded to the combine vectors logic 406 (Fig. 4). Processing continues with step 518.

At step 518, the ingress port number corresponding to the ingress port 108 (Fig. 1) at which the IP Multicast data packet 106 (Fig. 1) was received is removed from the bridge vector 416 (Fig. 4). Processing continues with step 519.

At step 519, the bridge vector 416 (Fig. 4) and a VLAN membership vector (not shown) are combined. Processing continues with step 520.

At step 520, the bridge vector 416 (Fig. 4) and the route vector 414 (Fig. 4) are combined by performing a logical OR function to provide a Multicast forward vector 120 (Fig. 4) for the IP Multicast group. Processing continues with step 522.

At step 522, the Multicast forward vector 422 is forwarded to the egress port_q logic 134 (Fig. 1).

If no matching entry is found in the CAM 400 (Fig. 4) for the IP Multicast data packet 106 (Fig. 1), that is, the IP Multicast group is unknown, the Multicast data packet 106 (Fig. 1) is forwarded to the host port (not shown) so that the host port may add an entry for the IP Multicast group in the CAM 400 (Fig. 4), Route Mapper 402 (Fig. 6) and Route Table 404 (Fig. 4). The Multicast forward vector 422 (Fig. 4) is selected so that the IP Multicast packet 106 is forwarded (Fig. 1) to the host port. The Multicast forward vector 422 (Fig. 4) is also selected so that the IP Multicast data packet 106 (Fig. 1) for the unknown IP Multicast group is bridged to the VLAN on which the IP Multicast data packet 106 (Fig. 1) was received.

Fig. 6 illustrates the egress Multicast forwarding logic 118 shown in Fig. 1. The Egress Multicast forwarding logic 118 (Fig. 1) includes MFI detect logic 602, editor logic 600 and an MFI table 604. The MFI table 604 includes an MFI entry 612.

5 As a copy of an ingress modified IP Multicast data packet 126 (Fig. 1) stored in memory 116 (Fig. 1) forwarded on data-out 124 (Fig. 1) is received by the egress IP Multicast processing logic 118, the MFI detect logic 602 detects the MFI 132 included in the ingress modified IP Multicast data packet 126 (Fig. 1) and forwards the MFI 132 to the MFI Table 604. The MFI 132 is an index to the MFI entry 612
10 in the MFI table 604 corresponding to the IP Multicast group. The MFI entry 612 determines to which egress ports 112 (Fig. 1) the first copy of the IP Multicast data packet 106 (Fig. 1) is to be forwarded. If a subsequent IP Multicast data packet 106 is to be forwarded on an egress port 110, a next MFI 132' is included in the MFI entry 612 in the MFI table 604. The next MFI 132' is forwarded to the MFI detect
15 logic 602 to select the next MFI entry 612. For example, a next MFI 132' is stored with the MFI entry 612 for egress modified IP Multicast data packet 112a (Fig. 1) to select the next MFI entry 612' for egress modified IP Multicast data packet 112b (Fig. 1).

20 The ingress modified IP Multicast data packet 126 (Fig. 1) is modified by the editor logic 600 dependent on Multicast parameters 610 included in the MFI entry 612. Multicast parameters 610 stored in the MFI entry 612 are forwarded to the editor logic 600. Thus, each ingress modified IP Multicast data packet 126 (Fig. 1) is edited dependent on the destination node 104a-f (Fig. 1) to which it is being forwarded. By providing an MFI entry 612 in the MFI table 604, the ingress
25 modified IP Multicast data packet 126 (Fig. 1) may be edited to provide an egress modified IP Multicast data packet 112 (Fig. 1) and forwarded at wire speed to each member of the IP Multicast group connected to an egress port 110a-c (Fig. 1).

Fig. 7 illustrates the MFI table 604 shown in Fig. 6 for a switch 100 with four ports as shown in Fig. 1. The MFI table 604 (Fig. 6) includes an IVID table

700 and a VLAN ID table 702. An MFI entry 612' includes an IVID entry 718 in the IVID table 700 and a VLAN ID entry 710 in the VLAN ID table 702. An IVID entry 718 in the IVID table 700 includes a port field 714a-d for each port connected to the switch 100 (Fig. 1). The switch 100 as shown in Fig. 1 includes four ports:
5 ingress port _1 108 and egress ports _1-3 110a-c.

Each port field 714 includes a Bridge/Route field 708, a Next field 706 and an IVID field 704. The IVID entry 718 also includes a next MFI field 716. The next MFI field stores a next MFI 132' if any next field 706a-d in the IVID entry 718 indicates that another egress modified IP Multicast data packet 112 (Fig. 1) is to be forwarded to a port connected to the switch 100. (Fig. 1) The IVID entry 718 as shown is for a switch 100 (Fig. 1) with four ports, as shown in Fig. 1. The number of port fields 714 is not limited to the four shown, a port field 714 is provided for each port connected to the switch 100 (Fig. 1). For example, for a switch with
10 sixty-four ports, the IVID entry 718 includes sixty four port fields 714.

The IVID field 704 stores an index to a VLAN ID entry 716 in the VLAN ID table 702 corresponding to the IVID field 704 if there is a valid VLAN ID entry 716 for the port. The VLAN ID entry 716 in the VLAN ID table 702 includes a VLAN ID field 710 and a Tag field 712.

In an alternate embodiment the IVID entry 718 in the IVID Table 700 may include the VLAN ID field 710 and tag field 712 stored in the VLAN ID entry 716, in which case the IVID field 704 would not be required. However, storing the IVID entry 718 and the VLAN ID entry 710 separately, that is, in the IVID table 700 and the VLAN ID table 702 reduces the total memory required to implement the MFI table 604 (Fig. 6) because every ingress modified IP Multicast data packet 126 may
20 not have a corresponding VLAN ID entry 716.

The Bridge/Route field 708 is implemented as a single bit. The state of the Bridge/Route field 708 determines whether the ingress modified IP Multicast data packet 126 (Fig. 1) is to be bridged or routed to a port. The Next field 706 is

implemented as a single bit. The state of the next field 706 determines whether another egress modified IP Multicast data packet 112 (Fig. 1) is to be forwarded on the port. The IVID field 704 provides an index to the VLAN ID table 702. The number of bits in the IVID field 704 is dependent on the size of the VLAN ID table 702. For example, for a VLAN ID Table 706 with 256 VLAN ID entries 716 the VLAN ID index 714 includes eight bits.

The VLAN ID 710 contains a valid VLAN ID 710 if the ingress modified IP Multicast Data packet 126 (Fig. 1) is to be routed to the port. If the Tag field 712 indicates that the egress modified IP Multicast data packet 112 (Fig. 1) is not tagged, the editor logic 600 (Fig. 6) deletes the tag 318b (Fig. 3B) from the VLAN ID 318 (Fig. 3B) in the Ethernet data link layer (L2) header 304 (Fig. 3B).

Having determined that a VLAN ID 710 is required and obtaining the VLAN ID 710 from the VLAN ID table 702, the Multicast parameters 610 (Fig. 6) are forwarded to the editor logic 600 (Fig. 6). The editor logic 600 (Fig. 6) uses the Multicast parameters 610 to modify the ingress modified IP Multicast data packet 126 (Fig. 6), and after modifying the editor logic 600 (Fig. 6) forwards the egress modified IP Multicast data packet 112 to the egress port 110 (Fig. 1).

If the Bridged/Routed field 708 in the port field 714 is set to bridge. The editor logic 600 (Fig. 6) adds or removes the VLAN ID 318 (Fig. 3C) and restores the destination address 314 (Fig. 3B) in the Ethernet data link layer (L2) header 304 (Fig. 3B) to its original value.

If the Bridged/Routed field 708 in the port field 714 is set to routed. The editor logic 600 (Fig. 6) overwrites the router's source address in the source address 314 (Fig. 3B) in the Ethernet data link layer (L2) header 304 (Fig. 3B), decrements the TTL 336 (Fig. 3C) and generates an IP check sum and stores it in header checksum 342 (Fig. 3C). The editor logic 600 (Fig. 6) also restores the IP destination address 346 (Fig. 3C) to the original value and adds the TCI 318b (Figure 3B) tag.

The egress Multicast forwarding logic 118 (Fig. 1) forwards the IP destination address 346 (Fig. 3C) the egress modified IP Multicast data packet 112a-f (Fig. 1) to the egress ports 110a-c (Fig. 1).

5 While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

CLAIMS

What is claimed is:

1. A switch comprising:

5 a memory storing a multicast data packet received from a port connected to the switch;

egress port forwarding logic, the egress port forwarding logic comprising:

a modification entry for each member of the multicast group associated with the multicast data packet; and

10 editing logic which modifies a copy of the multicast data packet received from the memory dependent on the modification entry.

2. The switch as claimed in Claim 1 further comprising:

15 ingress port forwarding logic which generates a multicast forward vector for the multicast data packet, the multicast forward vector indicating to which ports the stored multicast data packet is to be forwarded.

3. The switch as claimed in Claim 2 further comprising:

port queue logic which stores a pointer to the multicast data packet stored in the memory in at least one port queue dependent on the multicast forward vector.

20 4. A switch as claimed in Claim 3 wherein the port queue logic removes the pointer to the multicast data packet stored in the port queue after the stored multicast data packet has been forwarded to all members of the IP Multicast group connected to the port associated with the port queue.

25 5. A switch as claimed in Claim 1 wherein the modification entry comprises:

a virtual LAN identification;

a pointer to a next modification entry;

a tag indicator; and

a forward type indicator.

6. A switch as claimed in Claim 5 wherein the pointer to next modification entry is an index to a next modification entry for modifying a next copy of the multicast data packet.

5 7. A switch as claimed in Claim 1 wherein the modification entry comprises:
a virtual LAN entry stored in a virtual LAN identification table; and
an intermediate entry stored in an intermediate entry stored in an
intermediate LAN identification table wherein the intermediate entry includes an
index to the virtual LAN entry.

10 8. A method for forwarding multicast data packets in a switch comprising the steps of:

15 storing in a memory, a multicast data packet received from an ingress port;
providing egress port forwarding logic, the egress port forwarding logic
including a modification entry for each member of the multicast group associated
with the multicast data packet and editing logic for modifying the multicast data
packet;

20 modifying in the egress port forwarding logic a copy of the multicast data
packet received from the memory dependent on the modification entry; and
forwarding by the egress port forwarding logic the modified copy of the
multicast data packet to an egress port in a next available egress port cycle time.

25 9. The method for forwarding multicast packets in a switch as claimed in Claim
8 further comprising the step of:

selecting a multicast forward vector for the multicast data packet dependent
on parameters included in the multicast data packet, the multicast forward vector
indicating to which egress ports the multicast data packet is to be forwarded.

10. The method for forwarding multicast packets in a switch as claimed in Claim
9 further comprising the step of:

storing, by the packet manager a pointer to the multicast data packet stored in the memory in a port queue associated with the egress port dependent on the multicast forward vector.

5 11. The method for forwarding multicast packets in a switch as claimed in Claim 10 further comprising the step of:

removing, by the packet manager the pointer to the multicast data packet stored in the port queue after all modified copies of the multicast data packet are transmitted to the egress port associated with the port queue.

10 12. The method for forwarding multicast packets in a switch as claimed in Claim 8 wherein the step of providing a modification entry comprises the step of:
providing a virtual LAN identification, a pointer to next entry, a tag indicator and a forward type indicator.

15 13. The method for forwarding multicast packets in a switch as claimed in Claim 12 wherein the pointer to next entry provides an index to a next modification entry storing the parameters for a second modified copy of the multicast data packet to be sent to the egress port.

20 14. The method for forwarding multicast packets in a switch as claimed in Claim 8 wherein the step of providing egress port forwarding logic further comprises the steps of:

providing a virtual LAN identification table; and
providing an intermediate LAN identification table for storing the multicast entry wherein the multicast entry includes an index to a location in the virtual LAN
25 identification table including a virtual LAN identification.

15. A switch comprising:

a memory storing a multicast data packet received from an ingress port connected to the switch;

egress port forwarding logic, the egress port forwarding logic comprising:

a modification entry for each member of the multicast group associated with the multicast data packet; and

means for modifying a copy of the multicast data packet received from the memory dependent on the modification entry.

5 16. The switch as claimed in Claim 15 further comprising:
 means for generating a multicast forward vector for the multicast data packet,
 the multicast forward vector indicating to which ports the stored multicast data
 packet is to be forwarded.

10 17. The switch as claimed in Claim 16 further comprising:
 means for storing a pointer to the multicast data packet stored in the memory
 in at least one port queue dependent on the multicast forward vector.

15 18. A switch as claimed in Claim 17 wherein the means for storing removes the
 pointer to the multicast data packet stored in the port queue after the multicast data
 packet has been transmitted to each member of the multicast group connected to the
 egress port associated with the port queue.

20 19. A switch as claimed in Claim 15 wherein the modification entry comprises:
 a virtual LAN identification;
 a pointer to next entry;
 a tag indicator; and
 a forward type indicator.

25 20. A switch as claimed in Claim 19 wherein the pointer to next entry provides
 an index to a next modification entry storing the parameters for a next modified
 copy of the multicast data packet to be forwarded to the egress port.

21. The switch as claimed in Claim 15 wherein the modification entry
comprises:

a virtual LAN entry stored in a virtual LAN identification table; and
an intermediate entry stored in an intermediate entry stored in an
intermediate LAN identification table wherein the intermediate entry includes an
index to the virtual LAN entry.

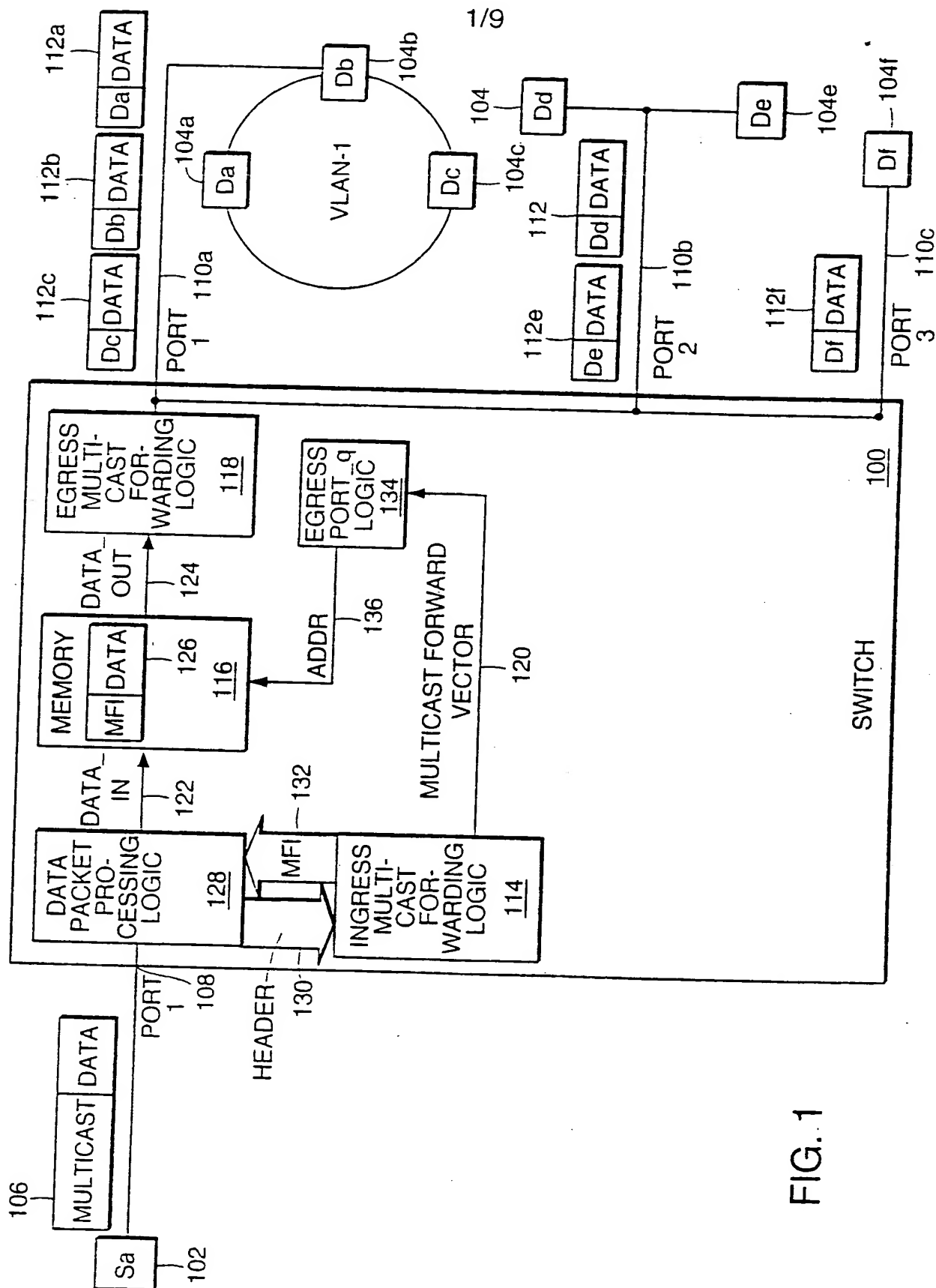


FIG. 1

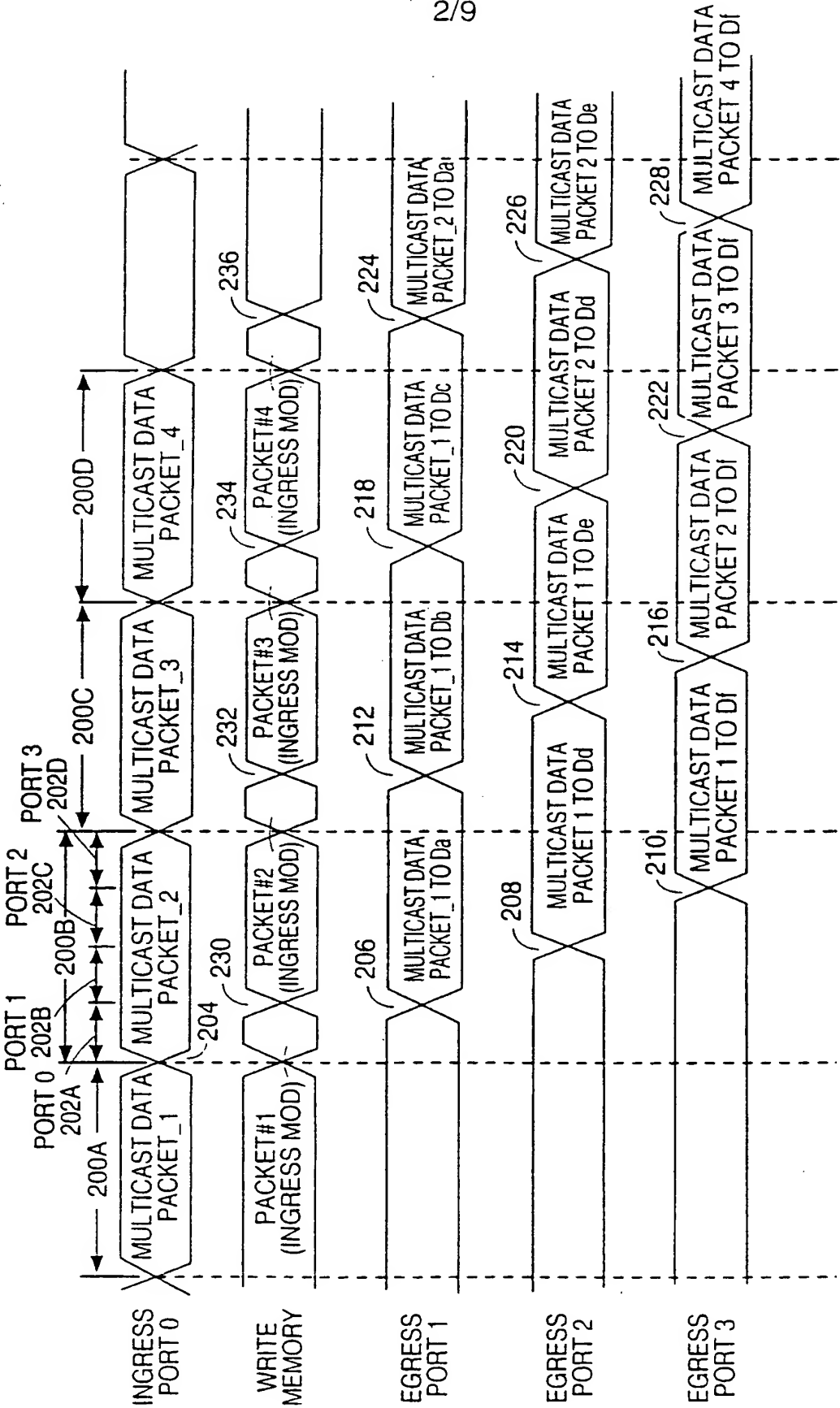
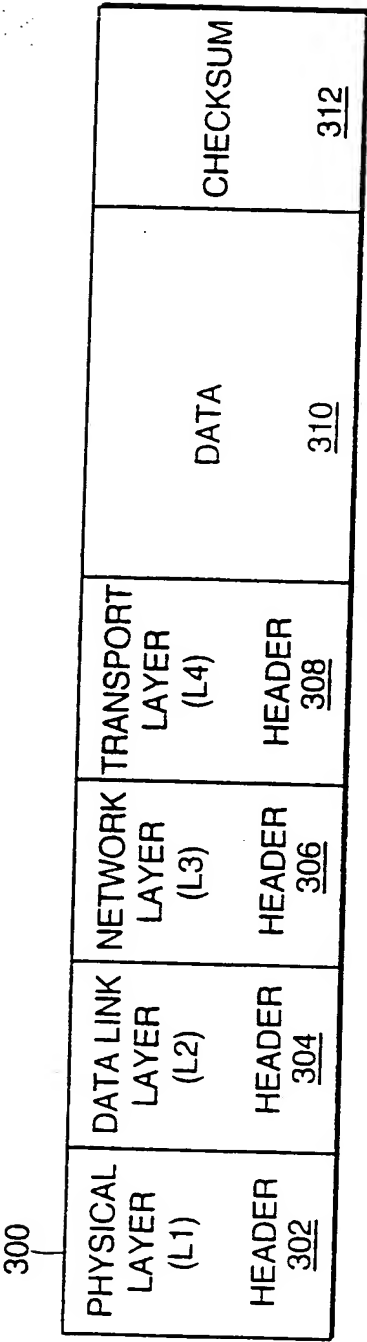
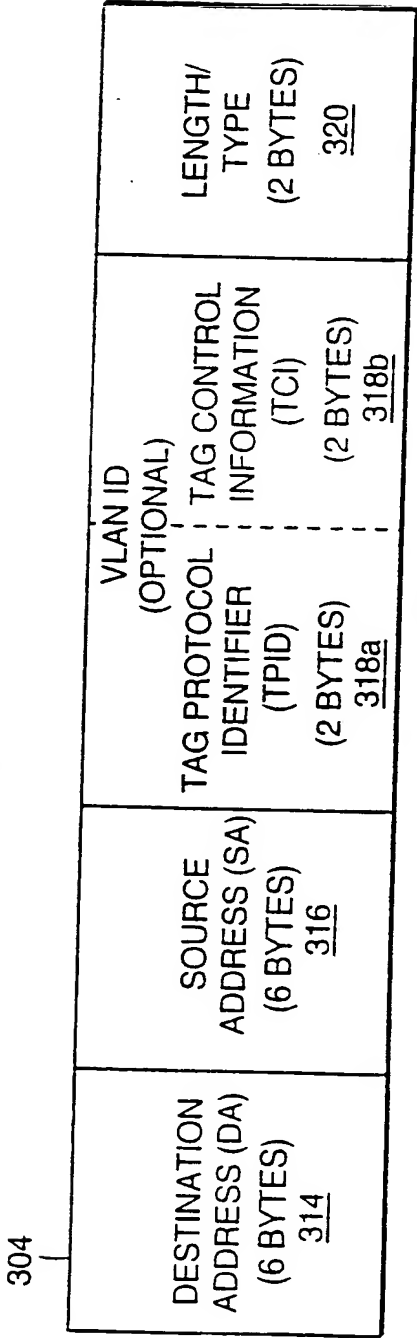


FIG. 2

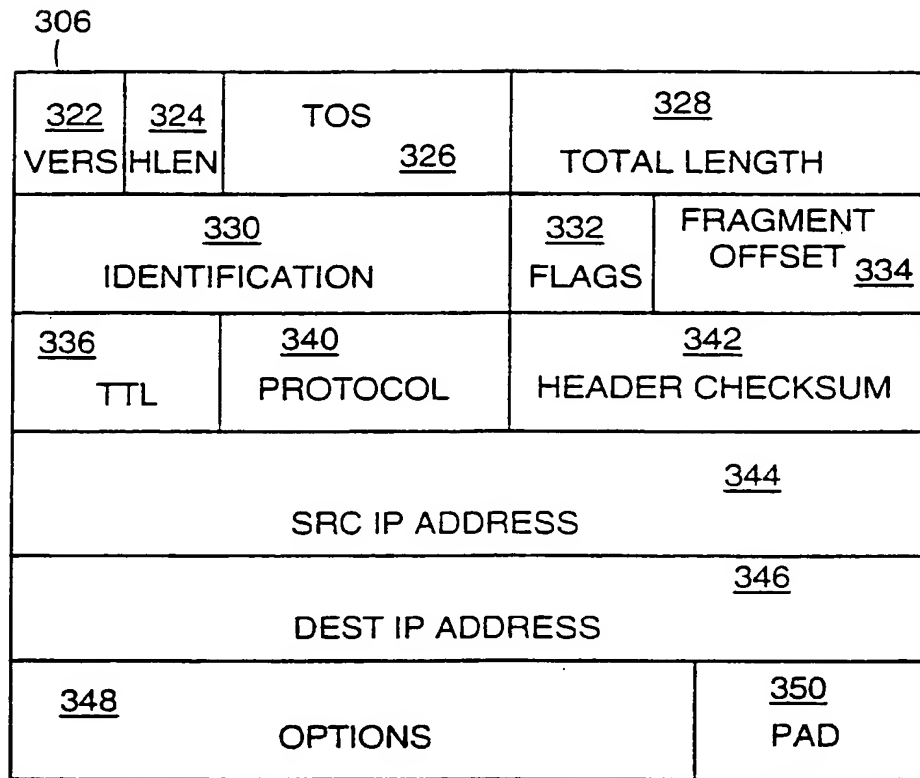


PRIOR ART
FIG. 3A

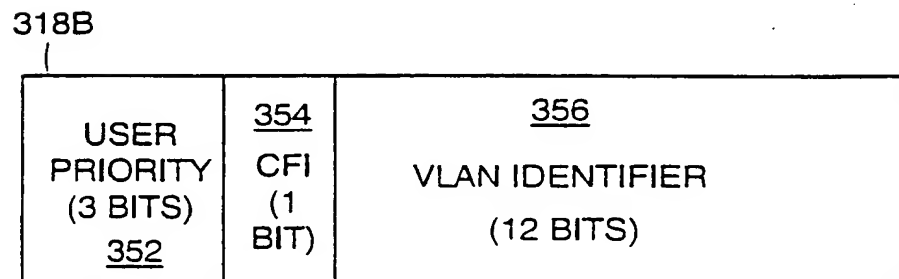


PRIOR ART
FIG. 3B

4/9



PRIOR ART
FIG. 3C



PRIOR ART
FIG. 3D

5/9

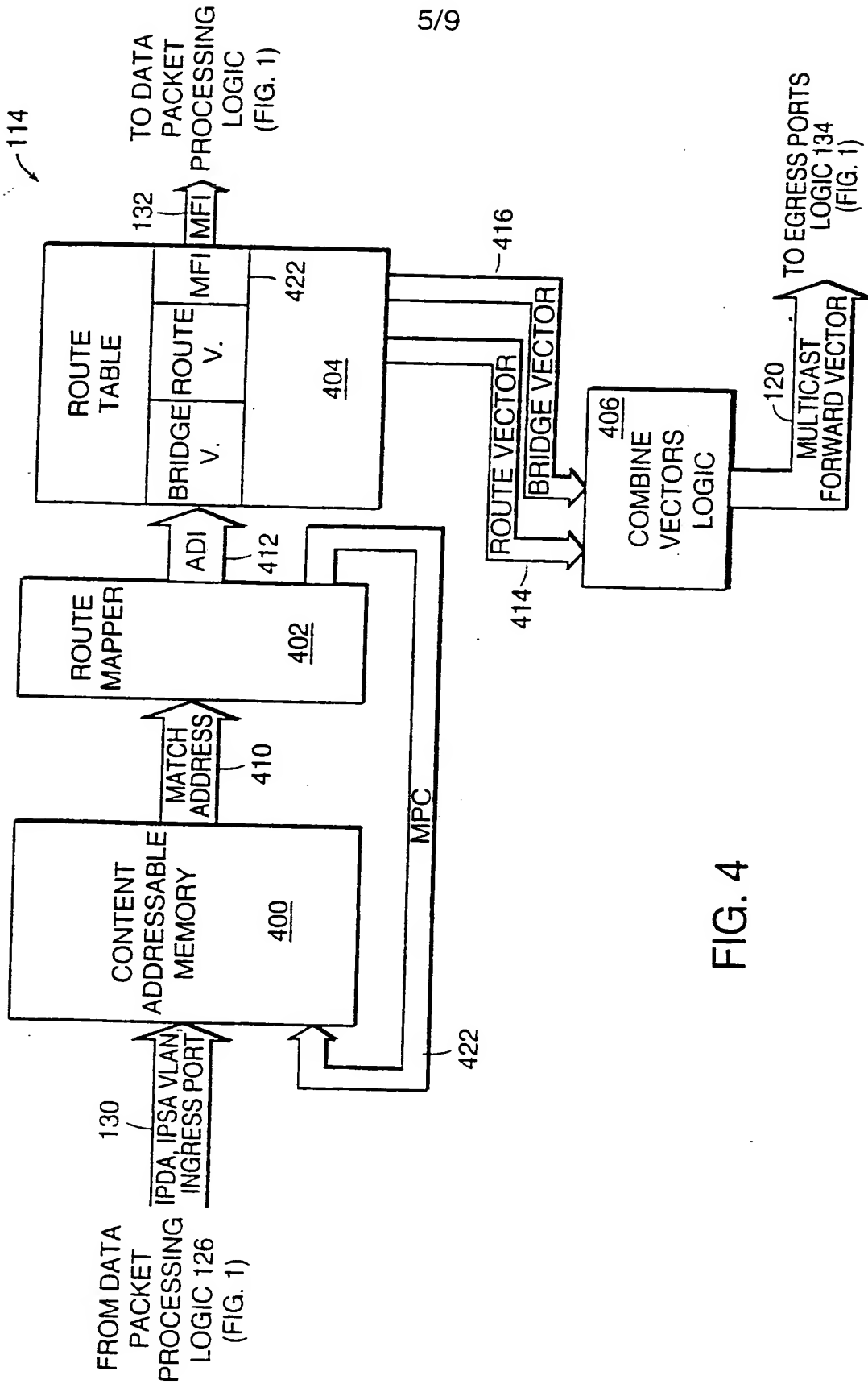


FIG. 4

6/9

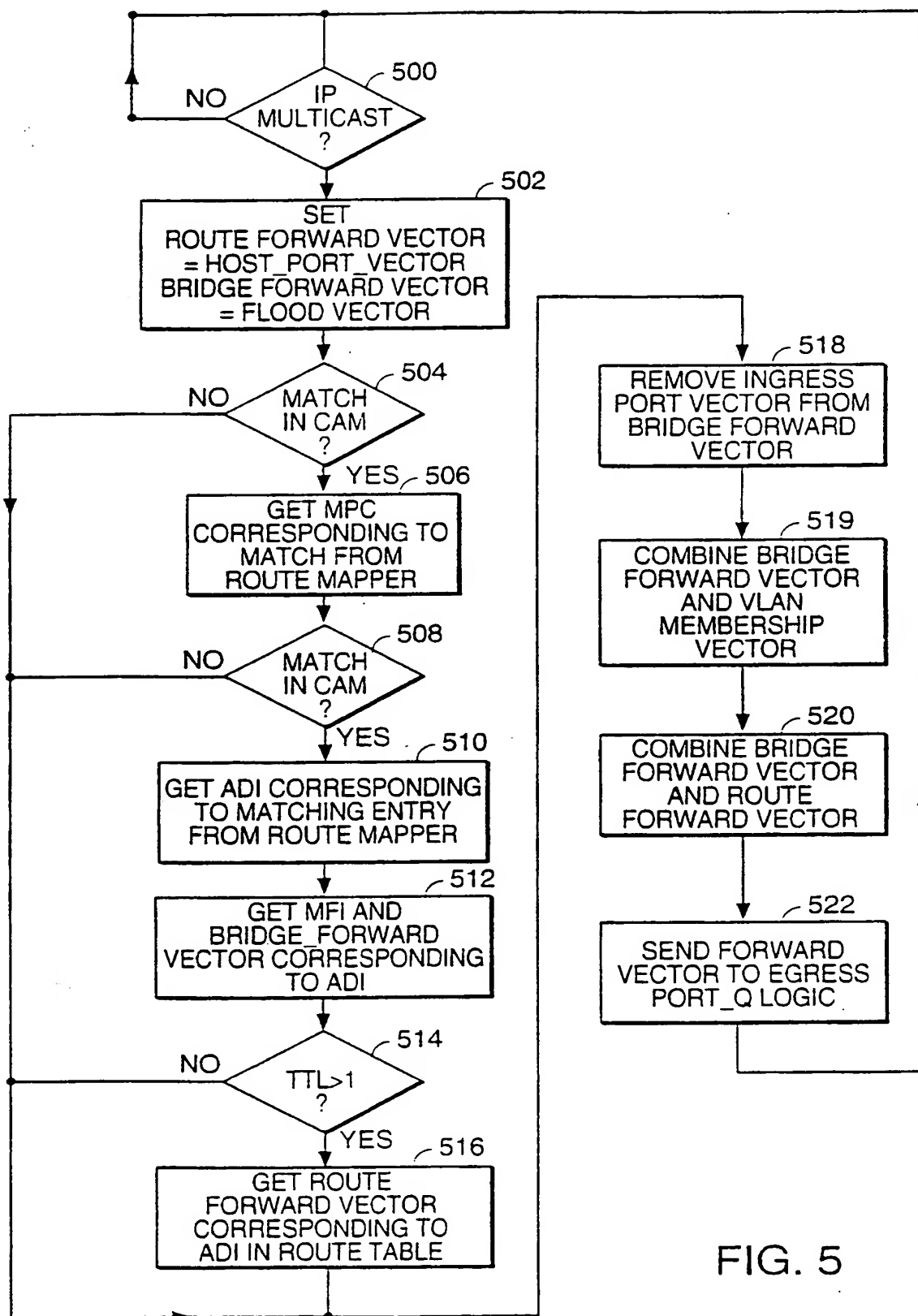


FIG. 5

118

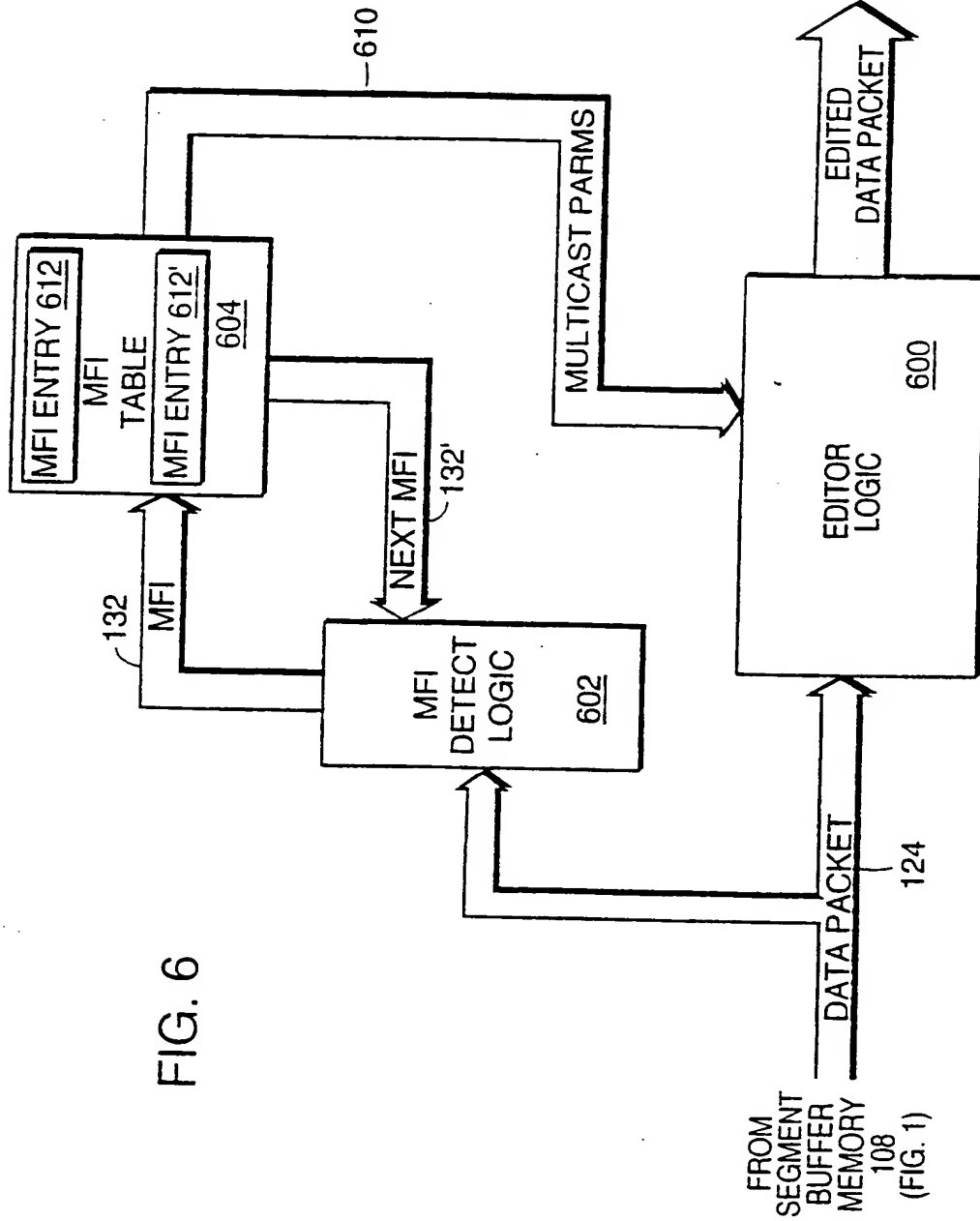
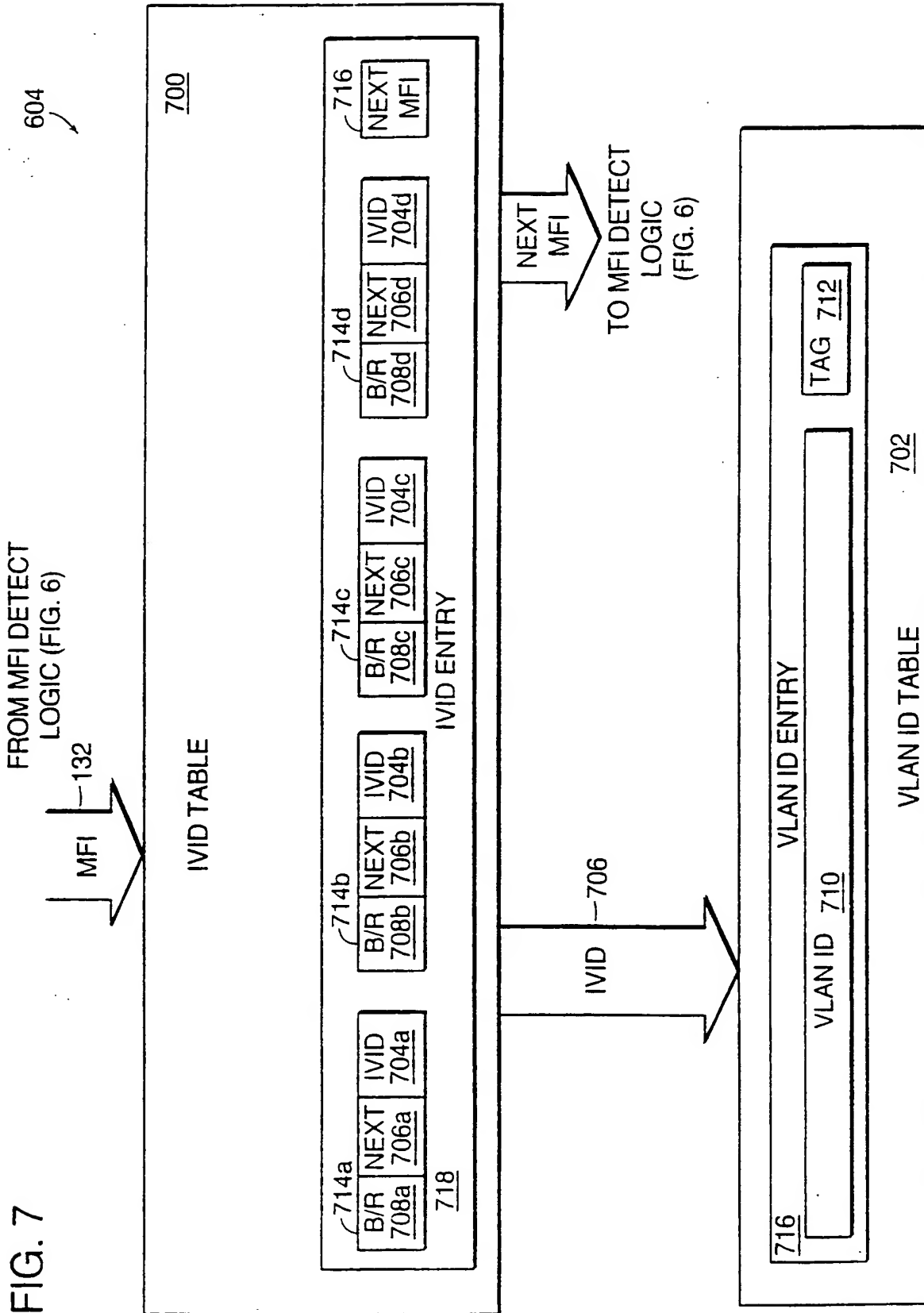
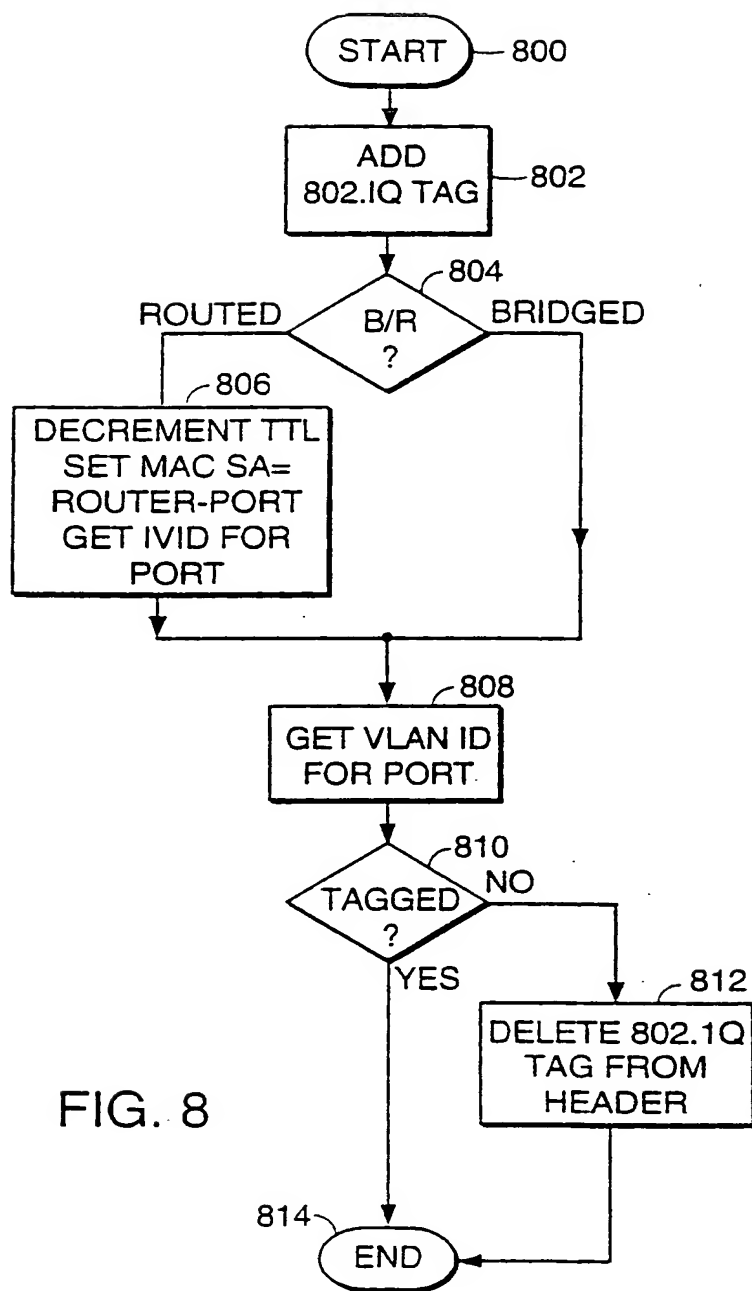


FIG. 6



9/9



(19) World Intellectual Property Organization
International Bureau



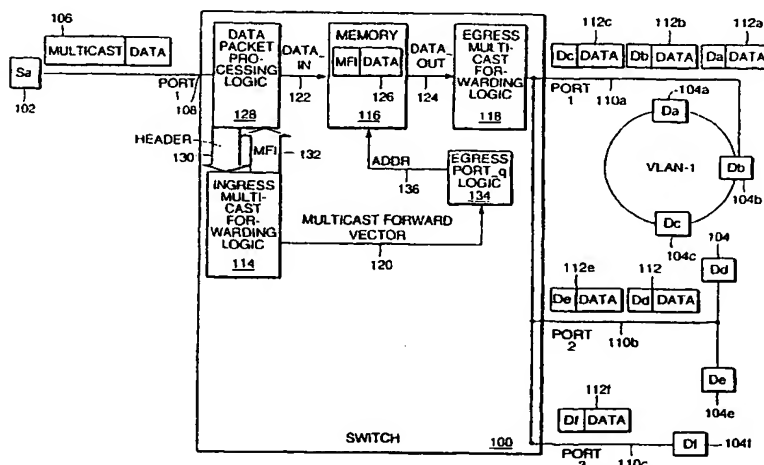
(43) International Publication Date
7 June 2001 (07.06.2001)

PCT

(10) International Publication Number
WO 01/41364 A3

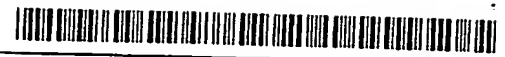
- (51) International Patent Classification⁷: **H04L 12/18**, 29/06. 12/46
- (21) International Application Number: **PCT/CA00/01420**
- (22) International Filing Date:
30 November 2000 (30.11.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/453,344 1 December 1999 (01.12.1999) US
- (71) Applicant (for all designated States except US): **MOSAID TECHNOLOGIES INCORPORATED [CA/CA]**; 11 Hines Road, Kanata, Ontario K2K 2X1 (CA).
- (72) Inventor: and
- (75) Inventor/Applicant (for US only): **BROWN, David, A.** [CA/CA]; 110 Allissia Crescent, Carp, Ontario K0A 1L0 (CA).
- (54) Title: **METHOD AND APPARATUS FOR IP MULTICASTING**
- (74) Agent: **PILLAY, Kevin**; Fasken Martineau DuMoulin LLP, Suite 4200, Box 20, Toronto-Dominion Centre, Toronto Dominion Bank Tower, Toronto, Ontario M5K 1N6 (CA).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— with international search report
- (88) Date of publication of the international search report:
6 December 2001

[Continued on next page]



(57) Abstract: In a switch including a plurality of ports, an IP Multicast packet arriving on an ingress port a copy of the receiver packet is forwarded to each member of the IP Multicast group at wire-speed. The packet is bridged once to a given egress port and may be routed multiple times out of the egress port. If multiple subnets exist on an egress port, each subnet that requires a copy of the packet will receive the packet with its VLAN ID included in the packet. The received IP Multicast packet for an IP Multicast group is stored in memory, a pointer to the location of the packet in memory is stored for each port to which a copy of the packet is to be forwarded. An IP Multicast forwarding entry is provided for the IP Multicast group. The forwarding entry includes a modification entry for each packet to be forwarded to the IP Multicast group. A copy of the stored packet is modified dependent on the modification entry and forwarded in the next available port cycle for the port.

WO 01/41364 A3



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

Inter. .onal Application No

PCT/CA 00/01420

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L12/18 H04L29/06 H04L12/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 963 552 A (JOO SANG-WAN ET AL) 5 October 1999 (1999-10-05)	1-3, 8-10, 15-17
A	column 3, line 22 -column 6, line 37	4-7, 11-14, 18-21
X	US 5 898 687 A (HARRIMAN GUY ET AL) 27 April 1999 (1999-04-27)	1-3, 8-10, 15-17
A	abstract column 3, line 46 -column 5, line 36 column 7, line 56 - line 67	4-7, 11-14, 18-21

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *G* document member of the same patent family

Date of the actual completion of the international search

4 July 2001

Date of mailing of the international search report

11/07/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Peeters, D

INTERNATIONAL SEARCH REPORT

International Application No

PCT/CA 00/01420

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>XIONG Y ET AL: "Analysis of multicast ATM switching networks using CRWR scheme" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 30, no. 8, 1 May 1998 (1998-05-01), pages 835-854, XP004122815 ISSN: 0169-7552 page 836, paragraph 2 -page 837; figures 1,2</p>	1-21
A	<p>"MULTICAST/BROADCAST MECHANISM FOR A SHARED BUFFER PACKET SWITCH" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 34, no. 10A, 1 March 1992 (1992-03-01), pages 464-465, XP000302372 ISSN: 0018-8689 the whole document</p>	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/CA 00/01420

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5963552 A	05-10-1999	NONE	
US 5898687 A	27-04-1999	NONE	

THIS PAGE BLANK (USPTO)